LA-UR- 02-~~[illegible]~~497

Approved for public release; distribution is unlimited.

c. \

Title:

Genetic Algorithm Dose Minimization for an Operational Layout

Author(s):

Steve McLawhorn, Drew E. Kornreich, Donald J. Dudziak, D-7

Submitted to:

the April 2002 Radiation Protection and Shielding Division Topical meeting in Santa Fe, New Mexico

Los Alamos
NATIONAL LABORATORY

GENETIC ALGORITHM DOSE MINIMIZATION FOR AN OPERATIONAL LAYOUT

Steve McLawhorn
Los Alamos National Laboratory
P. O. Box 1663
Los Alamos, New Mexico, 87545
505-665-4027
smclwhrn@lanl.gov

Drew Kornreich
Los Alamos National Laboratory
P. O. Box 1663
Los Alamos, New Mexico, 87545
505-667-2095
drewek@lanl.gov

Don Dudziak
North Carolina State University
Campus Box 7909, NCSU
Raleigh, North Carolina, 27695
919-515-1465
dudziak@ncsu.edu

SUMMARY
In an effort to reduce the dose to operating technicians performing fixed-time procedures on encapsulated source material, a program has been developed to optimize the layout of workstations within a facility by use of a genetic algorithm. Taking into account the sources present at each station and the time required to complete each procedure, the program utilizes a point-kernel dose calculation tool for dose estimates. The genetic algorithm driver employs the dose calculation code as a cost function to determine the optimal spatial arrangement of workstations to minimize the total worker dose.

I. BACKGROUND

Genetic algorithms[1,2] are algorithms applied to problems of numerical optimization. The processes employed by such algorithms are developed metaphorically from the fundamental principles of genetics and natural selection. The natural operations that the genetic algorithm mimics are straightforward and the procedural simplicity of the method makes it broadly applicable. They are robust in their search capability and typically converge quickly on solutions to problems with many parameters.

Unlike other optimization techniques, genetic algorithms begin with a population of possible solutions rather than a single-point guess. This initial population is chosen at random typically by use of a random number generator, and is therefore spread throughout the solution space to be searched. Each possible solution in the population is typically represented as a string of binary digits that together denote guesses for all parameters in the problem being analyzed. In this way the string that represents the set of parameter values for a given individual solution is analogous to the chromosome of an organism in nature. For example, in a simple one-parameter problem with an initial population of three, its representative strings might be 1100, 0101, and 0010 corresponding to guesses of 12, 5, and 2 respectively.

With the initial population generated and encoded, a selection process analogous to natural selection is performed. The problem to be analyzed, also referred to as a cost function, is solved using the parameter values of each individual string. Then, based on the criteria of the search, that is, maximization or minimization of the cost function, the individual solutions are assessed. At this point the more fit individuals are saved for mating and the less fit individuals are discarded from the population. This is analogous to the process in natural selection where by weaker individuals perish while stronger individuals survive to reproduce.

With the most-fit individuals selected, a crossover process analogous to breeding is carried out. Schemes for performing this process with representative strings range from simple to complex, that is, from single-point crossover in which strings are always broken and recombined at a specified point to processes where strings are broken and recombined at many points chosen by a random number generator. However, all involve swapping binary bits between individuals to create a new distinct individual that carries some attributes of both parents. As an example of a simple case of single-point crossover, consider a case in which the parent strings are divided in the center and all bits to the right of this point are swapped. Expanding upon the first example, suppose that out of the initial population of three individuals the strings representing the parameter values 12 and 5 were determined to be the most fit. Now, all that is needed to create a new generation of possible solutions is to cross the strings 1100 and 0101. This yields two offspring 1101 and 0100 corresponding to guesses of 13 and 4 respectively.

At this point a new and presumably more fit set of offspring solutions form the population of the next generation. However, to ensure that the most-fit individual of the previous generation, and perhaps of the current generation as well, is not lost this individual is carried forward into the next generation in a process referred to as elitist reproduction. If for instance the most-fit individual in the initial example population has been determined to be 12 then the next generation would include 12 and the two new offspring 13 and 4.

To complete the analogy to natural selection and prevent the algorithm from stalling in a local optimum, a mutation process is employed. Within randomly selected individuals, certain bits are chosen at random and flipped to create a mutated string. This process is applied very sparingly to prevent loss of integrity among the individuals of the population between generations, but the occasionally highly fit mutation can force the algorithm out of a local optimum.

Through successive generations this process will discover the global optimum with the advantages of converging quickly and being unlikely to become stuck in local optima.

II. APPROACH – DOSE MINIMIZATION

Implementing the genetic algorithm approach to optimizing the spatial layout of sources within the facility begins with defining the problem. The sources may be placed at any spatial point inside the facility limited only by the proportions of the room, and the dimensions of the workspace assigned to each source. The operations carried out at each workspace are allotted a specific amount of time for completion and take place in a predetermined order. The goal in arranging the sources throughout the room is to minimize the total worker dose accumulated during the process. A sample portion of the flowsheet is shown in Figure 1.

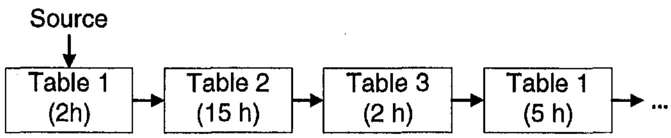

Figure 1. Sample operational flowsheet.

The program created to perform the optimization is comprised of two separate programs: the genetic algorithm and the dose estimation program that acts as the cost function. The variables sought by the genetic algorithm are the x and y coordinates for each source inside the boundaries of the facility. The fitness of each trial solution set is determined by the total dose calculated.

The time constraint at each workstation is expressed by weighting the dose from each station appropriately in summing the total dose. The total dose value is then supplied to the genetic algorithm to be used as the fitness value in determining the contents of the next generation of trial solutions. In this manner a solution of source positions yielding the lowest total worker dose will be determined.

III. TESTING

A. Genetic Algorithm

The genetic algorithm was tested to ensure it functioned properly. Some of these tests were standard for testing genetic algorithms. Other tests were created to solve problems based on cost functions similar to those that would be encountered in the actual facility optimization problem. The solutions to these simple problems were intuitively known. The problems that were created all consisted of arranging points within squares and boxes while using different features and modes of the genetic algorithm.

The DeJong test bed functions[3] were employed first in assessing the proper function of the genetic algorithm. These functions are considered to be one of the standard tools for testing genetic algorithms. The test bed consists of five functions all of which have many local optima in various configurations and are designed specifically to confuse search algorithms. These functions were run with satisfactory results.

The algorithm was next tested using a two-dimensional problem with 4 parameters. In this problem there were two points that were to be arranged inside a square in such a manner as to maximize the linear distance between the points within the space. The solution to this problem is intuitively known with the correct solution being to place the points at opposing corners. The test yielded a satisfactory result.

The same problem was then run as a minimization problem in which the program was required to minimize the linear distance between the points. In this case all that is relevant is that the program places the points as close together as possible. The positions of the points inside the square were otherwise unimportant. Again, the test yielded adequate results.

Similar problems of a more complex nature were created and used to further test the efficiency and function of the genetic algorithm. These included a three-dimensional (3-D) problem with six parameters involving the arrangement of three points inside a box, a 3-D problem with 10 parameters requiring the arrangement of 5 points, and a 3-D problem with twelve parameters constituting the placement of 6 points. All of these problems were run as both maximization and minimization problems with satisfactory results that agreed with the intuitive solutions. From these sample problems the genetic algorithm appeared to yield adequate solutions and typically converged quickly. All calculations were performed on a PC with a 450 MHz Pentium processor with 264MB RAM.

B. Dose Calculation

Dose estimates are obtained from a point-kernel photon-dose and diffusion-theory neutron-dose calculation tool called *Pandemonium*.[4] Pandemonium© uses a Visio™ interface to draw a two-dimensional geometry that contains sources, detectors, hydrogenous shields (and persons), and gloveboxes. When an item is placed in the Visio™ drawing page, a window opens where specific information for that item is entered. For example, when a glovebox is placed in the drawing, the thickness of the lead and steel is requested. When a source is placed in the drawing, isotopic, density, shielding, and material form information is requested. Once the geometry is constructed, the data is exported to the dose calculator.

Neutron dose is calculated according to

$$D_n = [S_{sf}h_E(E) + S_{(\alpha,n)}h_E(E)]\phi(R+a)e^{-0.15\Delta}, \qquad (1)$$

where the flux is calculated from diffusion theory at a distance a from the surface of a spherical source of radius R, the exponential term accounts for thermalization through hydrogenous material of total thickness Δ, the spontaneous fission and (α,n) source strengths are calculated according to material form and isotopics, and h_E is the fluence-to-dose conversion factor. The photon dose is calculated using a summation over energy groups of the photon flux (obtained using formulas for self-absorbing spherical sources and ANSI attenuation coefficients and buildup factors) times the appropriate fluence-to-dose factors.

Because the genetic algorithm effectively supplies the geometric information to Pandemonium©, the Visio™ interface is bypassed and the only component of Pandemonium© used is the actual dose calculation module.

C. Coupled Dose and GA Program

Once the Pandemonium© program was coupled to the genetic algorithm as a cost function, the new program was tested using a number of simple problems with intuitively known solutions. These problems include a two-source problem in a square 30'x30' room with detectors always at 36" from the sources. The detectors represent the positions at which workers will be stationed. Thus, the dose calculated at a detector yields the dose to a worker from all sources in the room. In this case the new code yielded a solution that placed the sources at opposite corners of the room. This is intuitively the correct answer. Also, an identical problem with five sources and detectors was run yielding a solution in which four of the sources were placed at the corners and one in the center. This is again intuitively the correct answer. Finally, a problem was run identical to the last but in a 60'X30' room. In this case the new code gave a solution with four of the sources at the corners and one placed at the center of one of the long walls. Again, this solution was intuitively correct.

The test problems employed did not take into account the time weighting of the source doses. A new problem with three sources was devised. In this problem the time spent at one of the sources was considerably more than the other two. In this case the program placed the highly weighted source at the far end of the room and the two sources of lesser weight in the opposing corners.

Testing the program with more complex scenarios whose solutions are not intuitive known posed a significant problem. It was necessary to find a method of comparing the solution sets obtained from the program to determine if the genetic algorithm was indeed converging on the optimal solution. In response a simple rendering tool was employed to graphically represent the dose fields of each solution set. Comparing the dose field images of multiple solution sets using the pixel densities in the images helps assess a particular solution. In some tests the solutions provided by the genetic algorithm appeared to be counter-intuitive, but upon inspection proved to be correct.

As an example, Figure 2 contains four identical time-weighted sources. One source is time weighted by a factor of 2.5; this source is located in the lower right corner in Figure 2, farthest from the other three sources as expected.

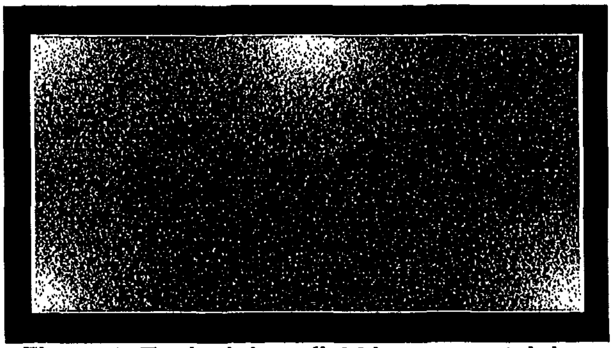

Figure 2. Typical dose field image containing four weighted sources in a 30'X60' facility.

IV. RESULTS AND CONCLUSIONS

The program was run using eight sources arranged within a 30'x60' area. Each source (all sources identical) was weighted to reflect the operational time associated with procedures performed on the given source. Workers were assumed to maintain a minimum distance of 36" from source material during operations. The dimensions of workstations and containers limited the source arrangement.

The results obtained were somewhat different from the arrangement that might be intuitively expected (see Table 1). The program placed two heavily weighted sources in proximity to one another. This would appear at first glance to be a mistake; however, when the total doses were calculated for this solution and compared to doses calculated for more instinctive solutions, the arrangement selected by the genetic algorithm generated a total dose approximately 10% lower than the next best solution. This solution provided a lower total dose by isolating two of the most heavily weighted sources from the remaining six workstations (i.e., it is protecting the low dose operations from the high-dose operations). This is a counterintuitive concept that might otherwise not be explored during planning.

Table 1. 8-Source Location Results

Source	Weight	Location
1	11/37	left center
2	16/37	top center
3	2/37	bottom center
8	4/37	center
4-7	1/37	corners

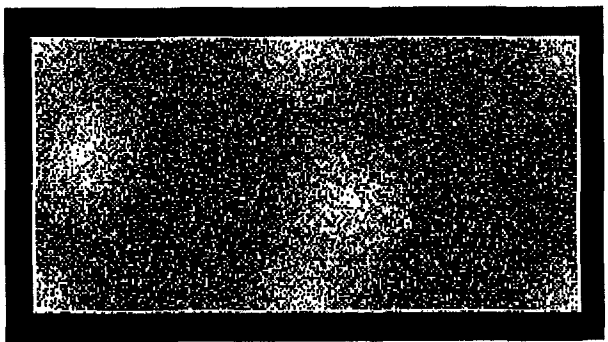

Figure 3. Dose field image of solution obtained by genetic algorithm.

The program was capable of obtaining what would appear to be the optimal solution to the problem presented. For the spatial arrangement of sources the program adequately provided a solution set that minimized the total worker dose during fixed time operations.

REFERENCES

1. R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, John Wiley and Sons, Inc., New York, New York (1998).

2. D. A. Coley, *An Introduction to Genetic Algorithms for Scientists and Engineers*, World Scientific, Singapore (1999).

3. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Mass. (1989).

4. D. E. Kornreich, D. E. Dooley, "A Two-Dimensional Point-Kernel Model for Dose Calculations in a Glovebox Array," *Transactions of the American Nuclear Society*, vol. 81, p. 249 (1999).